

Package ‘PALM’

November 6, 2025

Type Package

Title Fast and reliable association discovery in large-scale microbiome studies and meta-analyses

Version 0.1.0

Author Zhoujingpeng Wei

Maintainer Zhoujingpeng Wei <zwei74@wisc.edu>

Description This R package implements a quasi-Poisson-based framework designed for robust, scalable, and reproducible identification of covariate-associated microbial features in large-scale microbiome association studies and meta-analyses.

Data 2025-11-06

Reference Wei et al. 'Fast and reliable association discovery in large-scale microbiome studies and meta-analyses using PALM'. Submitted.

License GPL (>= 3) | file LICENSE

LinkingTo Rcpp, RcppArmadillo

Imports MASS (>= 7.3-60),
metafor (>= 4.8.0),
abess (>= 0.4.10),
dplyr (>= 1.1.4),
brglm2 (>= 0.9.2),
Rcpp (>= 1.0.14),
RcppArmadillo (>= 14.2.2-1)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr,
rmarkdown

Depends R (>= 4.3.0)

VignetteBuilder knitr

Contents

CRC_data	2
palm	2
palm.get.summary	5
palm.meta.summary	6
palm.null.model	7

CRC_data	<i>Datasets from two metagenomics studies of colorectal cancer (CRC)</i>
----------	--

Description

The "CRC_data" includes "CRC_abd" and "CRC_meta". "CRC_abd" is a sample-by-feature matrix of relative abundance counts from the two studies including 267 species under order "Clostridiales". The "CRC_meta" is a data frame including the sample-level variables from the two studies.

Usage

```
data(CRC_data)
```

Format

An object of class list of length 2.

Source

<https://github.com/zellerlab/crc_meta>

References

Wirbel, Jakob et al. Nat Med. 2019 Apr;25(4):679-689.

Examples

```
library("PALM")
data("CRC_data")
CRC_abd <- CRC_data$CRC_abd
CRC_meta <- CRC_data$CRC_meta
```

palm	<i>Meta-analysis for feature-level association testing for each covariate of interest</i>
------	---

Description

Implements the PALM framework for both single-study association analysis and meta-analysis across multiple studies. This function conducts feature-level association testing, evaluating one covariate of interest at a time.

Usage

```
palm(
  rel.abd,
  covariate.interest,
  covariate.adjust = NULL,
  cluster = NULL,
  depth = NULL,
  depth.filter = 0,
  prev.filter = 0.1,
  p.adjust.method = "fdr",
  meta.method = "EE",
  correct = "median"
)
```

Arguments

- | | |
|---------------------------------|--|
| <code>rel.abd</code> | For a single association study, provide a matrix of relative abundance counts, with sample IDs as row names and microbial feature IDs as column names. The matrix must not contain any missing values. For a meta-analysis, provide a list of such matrices, where each element corresponds to one study, and the name of each element represents the study ID. |
| <code>covariate.interest</code> | For a single association study, provide a matrix where rows represent samples and columns represent numeric covariates of interest. The order of samples must match the order in <code>rel.abd</code> , and the matrix must not contain any missing values. For a meta-analysis, provide a list of such matrices, where each element corresponds to one study, and the name of each element represents the study ID. The set of covariates may differ across studies. |
| <code>covariate.adjust</code> | For a single association study, provide a data frame where rows represent samples and columns represent covariates to be adjusted for in the model. Covariates must be either factors or numeric vectors. The order of samples must match the order in <code>rel.abd</code> , and the data frame must not contain missing values. For a meta-analysis, provide a list of such data frames, where each element corresponds to one study, and the name of each element represents the study ID. The set of covariates may differ across studies. The default is <code>NULL</code> , meaning no covariates are adjusted. |
| <code>cluster</code> | For a single association study with correlated samples, provide a vector defining the sample clusters. The order of samples in the vector must match the order in <code>rel.abd</code> . For example, the values may represent subject IDs if each subject has multiple correlated samples (e.g., in a longitudinal study). For a meta-analysis, provide a list of such vectors, where each element pertains to one study with correlated samples, and the name of each element represents the study ID. The order of samples in each vector must match the order in <code>rel.abd</code> for the corresponding study. The default is <code>NULL</code> , assuming all samples across all studies are independent. |
| <code>depth</code> | For a single association study, provide a vector representing the sequencing depth for each sample. The length of the vector must match the number of samples, and the order must align with <code>rel.abd</code> . For a meta-analysis, provide a list of such vectors, where each element corresponds to one study, and the name of each element represents the study ID. The default is <code>NULL</code> , in which case the sequencing depth is calculated as the row sums of <code>rel.abd</code> . |

<code>depth.filter</code>	A cutoff value used to remove samples with sequencing depth less than or equal to the cutoff. Default is 0.
<code>prev.filter</code>	A cutoff value to remove microbial features with prevalence (proportion of nonzero observations) less than or equal to the cutoff. This cutoff is applied to each study, so a feature may be removed in a subset of studies. The cutoff value must be in the range 0–1. Default is 0.1.
<code>p.adjust.method</code>	Character string specifying the multiple testing correction method. Options include "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", and "none". See <code>?stats::p.adjust</code> for more details.
<code>meta.method</code>	Character string specifying whether an equal-effects or random-effects model should be fitted. An equal-effects model is fitted when <code>method = "EE"</code> . A random-effects model is fitted by setting <code>method</code> to one of the following: "DL", "HE", "HS", "Hsk", "SJ", "ML", "REML", "EB", "PM", "GENQ", "PMM", or "GENQM". The default is "EE". See <code>?metafor::rma</code> for more details.
<code>correct</code>	Compositional effect correction method. The default is "median". If "median", the compositional effect is corrected using the median of the estimates. If "tune", the compositional effect is corrected using a data-driven tuned value. If NULL, no compositional correction is applied (i.e., the analysis uses relative-abundance summary statistics directly and outputs RA-level summary statistics). Recommendation: For sparse signals ($\leq 20\%$ differential AA features), use "median". For moderately dense signals ($> 20\%$ and $\leq 50\%$), "tune" tends to be more accurate. Both methods assume that the proportion of differential AA features is $\leq 50\%$.

Value

A list containing one component for each covariate of interest.

For single-study analysis, the component includes:

<code>palm_single</code>	A data frame containing association effect estimates, standard errors, p-values, q-values, and summary statistics.
--------------------------	--

For meta-analysis across multiple studies, the component includes:

<code>palm_meta</code>	A data frame containing overall association effect estimates, standard errors, p-values, and q-values for testing overall effects; p-values and q-values for testing cross-study heterogeneity; and summary statistics (association effect estimates and standard errors) for individual studies.
------------------------	---

Author(s)

Zhoujingpeng Wei <zwei74@wisc.edu>

See Also

[palm.get.summary](#), [palm.null.model](#), [palm.meta.summary](#)

Examples

```
library(PALM)
data("CRC_data", package = "PALM")
CRC_abd <- CRC_data$CRC_abd
```

```

CRC_meta <- CRC_data$CRC_meta

##### Generate summary statistics #####
rel.abd <- list()
covariate.interest <- list()
for (d in unique(CRC_meta$Study)) {
  rel.abd[[d]] <- CRC_abd[CRC_meta$Sample_ID[CRC_meta$Study == d], ]
  disease <- as.numeric(CRC_meta$Group[CRC_meta$Study == d] == "CRC")
  names(disease) <- CRC_meta$Sample_ID[CRC_meta$Study == d]
  covariate.interest[[d]] <- matrix(disease, ncol = 1,
                                   dimnames = list(names(disease), "disease"))
}

meta.result <- palm(rel.abd = rel.abd, covariate.interest = covariate.interest)

```

palm.get.summary

*Generate PALM summary statistics for individual studies***Description**

This function takes the output object from the `palm.null.model` function and constructs summary statistics for the covariates of interest in each study. The resulting output can be directly used as input to the `palm.meta.summary` function to perform meta-analysis.

Usage

```

palm.get.summary(
  null.obj,
  covariate.interest,
  cluster = NULL,
  correct = "median"
)

```

Arguments

<code>null.obj</code>	The output object from the <code>palm.null.model</code> function.
<code>...</code>	Additional arguments passed to <code>palm</code> . In <code>covariate.interest</code> and <code>cluster</code> , the order of samples must match the order in <code>rel.abd</code> used in the <code>palm.null.model</code> function.

Value

A list containing one component per study. Each component includes the following elements:

<code>est</code>	A matrix containing the association effect estimates for the study. Rows correspond to microbial feature IDs and columns correspond to covariate IDs.
<code>stderr</code>	A matrix containing the standard errors of the association effect estimates for the study. Rows correspond to microbial feature IDs and columns correspond to covariate IDs.
<code>n</code>	The sample size for the study.

See Also

[palm.null.model](#), [palm.meta.summary](#), [palm](#)

Examples

```
library(PALM)
data("CRC_data", package = "PALM")
CRC_abd <- CRC_data$CRC_abd
CRC_meta <- CRC_data$CRC_meta

##### Generate summary statistics #####
rel.abd <- list()
covariate.interest <- list()
for (d in unique(CRC_meta$Study)) {
  rel.abd[[d]] <- CRC_abd[CRC_meta$Sample_ID[CRC_meta$Study == d], ]
  disease <- as.numeric(CRC_meta$Group[CRC_meta$Study == d] == "CRC")
  names(disease) <- CRC_meta$Sample_ID[CRC_meta$Study == d]
  covariate.interest[[d]] <- matrix(disease, ncol = 1,
                                   dimnames = list(names(disease), "disease"))
}

null.obj <- palm.null.model(rel.abd = rel.abd)
summary.stats <- palm.get.summary(null.obj = null.obj,
                                 covariate.interest = covariate.interest)
```

palm.meta.summary

Meta-analyze summary statistics across studies

Description

This function takes the summary statistics output from the `palm.get.summary` function and combines the results across studies to perform feature-level association testing for each covariate of interest.

Usage

```
palm.meta.summary(summary.stats, p.adjust.method = "fdr", meta.method = "EE")
```

Arguments

`summary.stats` The output object from the `palm.get.summary` function.
`...` Additional arguments passed to `palm`.

Value

An object with the same structure as the output of the `palm` function when performing meta-analysis.

See Also

[palm.null.model](#), [palm.get.summary](#), [palm](#)

Examples

```
library(PALM)
data("CRC_data", package = "PALM")
CRC_abd <- CRC_data$CRC_abd
CRC_meta <- CRC_data$CRC_meta

##### Generate summary statistics #####
rel.abd <- list()
covariate.interest <- list()
for (d in unique(CRC_meta$Study)) {
  rel.abd[[d]] <- CRC_abd[CRC_meta$Sample_ID[CRC_meta$Study == d], ]
  disease <- as.numeric(CRC_meta$Group[CRC_meta$Study == d] == "CRC")
  names(disease) <- CRC_meta$Sample_ID[CRC_meta$Study == d]
  covariate.interest[[d]] <- matrix(disease, ncol = 1,
                                   dimnames = list(names(disease), "disease"))
}

null.obj <- palm.null.model(rel.abd = rel.abd)
summary.stats <- palm.get.summary(null.obj = null.obj,
                                 covariate.interest = covariate.interest)

##### Meta-analysis #####
meta.result <- palm.meta.summary(summary.stats = summary.stats)
```

palm.null.model

Get information from the null model

Description

The `palm.null.model` function computes the estimated mean microbial feature proportions and residuals of the relative abundances under the null model for each study. This function does not require the `covariate.interest` information. The output of this function is used as input for the `palm.get.summary` function to construct summary statistics for the covariates of interest in each study.

Usage

```
palm.null.model(
  rel.abd,
  covariate.adjust = NULL,
  depth = NULL,
  depth.filter = 0,
  prev.filter = 0.1
)
```

Arguments

... Additional arguments passed to `palm`.

Value

A list containing one component per study. Each component includes the following elements:

<code>Y_I</code>	A matrix of predicted abundances.
<code>Y_R</code>	A matrix of abundance residuals.
<code>Z</code>	The cleaned and formatted covariate.adjust.
<code>rm.sample.idx</code>	Indices of the samples removed from the study.

See Also

[palm.get.summary](#), [palm.meta.summary](#), [palm](#)

Examples

```
library(PALM)
data("CRC_data", package = "PALM")
CRC_abd <- CRC_data$CRC_abd
CRC_meta <- CRC_data$CRC_meta

##### Generate summary statistics #####
rel.abd <- list()
for (d in unique(CRC_meta$Study)) {
  rel.abd[[d]] <- CRC_abd[CRC_meta$Sample_ID[CRC_meta$Study == d], ]
}

null.obj <- palm.null.model(rel.abd = rel.abd)
```